Boolean Difference Calculus and Fault Finding

I. S. Reed
Electrical Engineering Department, University of Southern California

In this article a method is devised for testing for a possible fault in a gate in a larger switching circuit, that does not require isolating the suspicious gate from the rest of the circuit. The techniques involve a Boolean difference calculus reminiscent of, but not identical to, ordinary difference calculus.

I. Switching Circuits and Gates

A switching circuit $f(x_1, x_2, \cdots, x_n)$ is a realization of the mapping $f|\mathbf{2}^{(n)} \to \mathbf{2}$ where $\mathbf{2}$ is the set of two elements, i.e., $\mathbf{2} = \{0, 1\}$, and $\mathbf{2}^{(n)}$ is cartesian products of $\mathbf{2}$, taken n times,

$$\mathbf{2}^{(n)} = \overbrace{\mathbf{2} \times \mathbf{2} \times \cdots \mathbf{2}}^{n \text{ times}}$$

Mapping f is called either a switching or Boolean function. Switching circuits are composed of a number of sub-switching circuits. These elementary switching circuits often occur in modules and are called gates.

A typical switching circuit is illustrated in Fig. 1. It happens that this circuit consists only of *nand* gates. In terms of its gates the circuit of Fig. 1 is the following set of five equations:

$$\left.\begin{aligned} f &= \overline{g_1 g_2} \\ g_1 &= \overline{x_1 g} \\ g_2 &= \overline{x_4 g} \\ g &= \overline{x_1 x_4 u} \\ u &= \overline{x_2 x_3} \end{aligned}\right\} \tag{1}$$

Variables x_1, x_2, x_3, x_4 are called the primary or input variables to the circuit. The outputs of the gates u, g, g_1, g_2 are known as internal or secondary variables, and finally variable f is the output of the circuit. The bars in Eq. (1) denote complementation, i.e., if $u \epsilon \mathbf{2}$, $\overline{u} = 1 \oplus u$ where $\oplus$ is sum, modulo 2.

Suppose f is a switching function of the n binary variables

$$x = (x_1, x_2, \cdots, x_n)$$

and u is an internal variable of the circuit. The dependence of f on both x and u is represented functionally as

$$y = f(x, u(x)) \tag{2}$$

where x denotes the binary variables $x_1, x_2, \cdots, x_n$. Necessary and sufficient conditions for f to be a dependent function of u are well known (Ref. 1).

To illustrate the notion of functional dependence, consider again the circuit of Fig. 1. The output

$$y = f(g_1(x), g_2(x)) \tag{3}$$

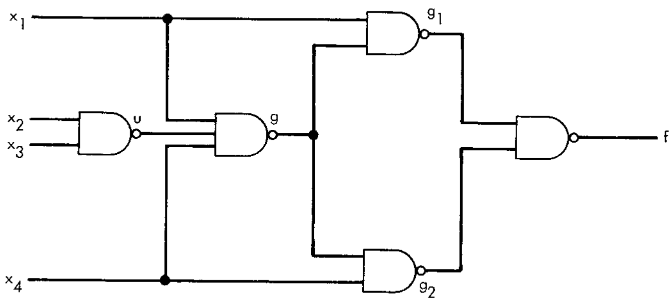

Fig. 1. A typical switching circuit

is a dependent function of both function $g_1(x)$ and function $g_2(x)$ where $x = (x_1, x_2, x_3, x_4)$. Similarly (see Eq. 1),

$$y_1 = g_1(x_1, g(x))$$

$$y_2 = g_2(x_2, g(x)) \tag{4}$$

are both dependent functions of a primary variable and of the same gate $g(x)$.

Equations (3) and (4) illustrate functionally another important concept. This is circuit fan-in and fan-out. Since $g_1(x)$ and $g_2(x)$ in (3) "feed" the single gate f, the outputs g_1 and g_2 are said to fan-in to f. In Eq. (4) gates g_1 and g_2 are both fed by the same gate g. Schematically Fig. 1 thus requires the output lines of g to fan out from gate g in order to simultaneously feed gates g_1 and g_2. The generalization of the concept of fan-in and fan-out to n gates is evident.

II. Faults and Their Detection

Let $y = f(x, u(x))$ be a function, dependent on an internal gate variable $u(x)$ where x denotes the set of primary variables $x_1, x_2, \cdots, x_n$. Our purpose is to devise a test for a possible error or fault in the physical realization of gate u.

If a gate such as u can be isolated from the rest of the circuit f, the testing of u would be a straightforward matter. However, today a subcircuit u of a digital integrated circuit usually cannot be disconnected without serious damage to the overall circuit f.

Assumption A. *The only terminals of the circuit f available for testing are the primary input variables $x_1, \cdots x_n$ and the output y.*

In order to impress both values 0 and 1, respectively, on u in a reliable manner with test equipment, it is convenient to assume there is only one or no fault in the circuit making up f. If u is the gate variable to be tested, one assumes the only possible error in the gates of f resides in the output of gate u. The above single-fault assumption is formalized as follows:

Assumption B. *Let $y = f(x, u(x))$ be a switching function dependent on gate function $u(x)$ where*

$$x = (x_1, x_2, \cdots, x_n).$$

If gate u is being tested, there is only one possible error or fault in the gates which physically compose f, and that is at the output of gate u.

Now consider the possible errors or faults which can occur at the output of gate u. The types of error possible in u are perhaps best illustrated by the symbol transition diagram of Shannon. Such a diagram is shown in Fig. 2. In Fig. 2 *impressed* values and *received* values correspond to transmitted symbols and received symbols, respectively, in the usual Shannon diagram for a binary (or two symbol) channel. $u(x)$ is the desired value or the value which the tester endeavored to impress upon the gate u, whereas $u^R(x)$ is the actual value at the output of gate u as a function of input configuration, $x \epsilon 2^n$. If one could disconnect the output of gate u from the rest of the circuit f and connect the output of gate u to a tester, then $u^R(x)$ would be the actual value received by the tester from gate u.

If the symbol or value k is impressed on gate u, the probability that the same value is received is given by q_k where $k = 0, 1$. The probability is $p_k = 1 - q_k$ that the wrong or opposite symbol is received.

In general, the probabilities in Fig. 2 allow for the possibility of intermittent errors. We are interested for the present only in the static or permanent type of error. These are the probability one type errors. Digital computer designers denote such errors as faults.

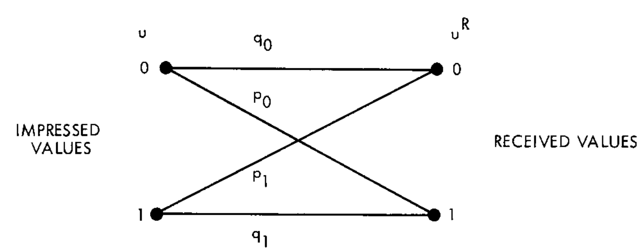

Fig. 2. Shannon error diagram for u

All possible probability one type of errors or faults are illustrated in Fig. 3. The type I fault is u stuck-at-zero. Type II fault is the u stuck-at-one. Finally, the type III error is u complemented.

Type III type errors occur quite frequently in final LSI (large scale integrated) circuit layouts. The bar for complementation is often omitted or inserted erroneously during the final conversion of switching circuit equations to planar circuits. Type I and II faults occur more usually as electronic faults.

We can now prove the following theorem:

THEOREM 1. Let $y = f(x, u(x))$ be a switching function dependent on a gate function $u(x)$ where $x = (x_1, x_2, \cdots, x_n)$. Under Assumptions A and B a single fault in u of any of the three types I, II, or III of Fig. 3 can be detected if and only if there exist two input test configurations x^0 and x^1 in 2^n such that

(1) $u(x^k) = k$

(2) $f(x^k, u(x^k)) \neq f(x^k, \overline{u}(x^k))$

for both $k = 0$ and $k = 1$.

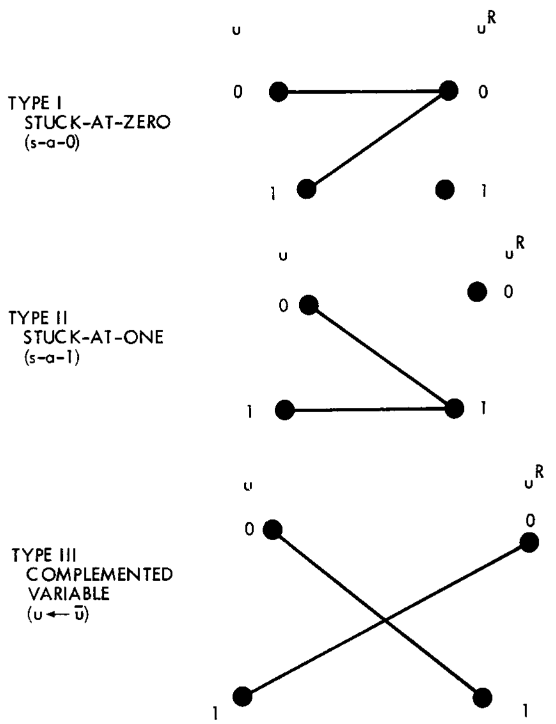

Fig. 3. Three types of faults

Proof: Suppose x^0 and x^1 exist such that (1) and (2) are true and u has a fault, e.g., a type I fault. From Fig. 3 we have

$$u^R(x^0) = u(x^0)$$
$$u^R(x^1) = \overline{u}(x^1)$$

for a type I fault. By Assumptions A and B only the output $y = f$ of the circuit is observable and the observed value of y is

$$y^R(x) = f(x, u^R(x)) \tag{5}$$

for input configuration $x \epsilon 2^n$. If x^0 and x^1 are test configurations, it is apparent that a fault in u can be detected if and only if

$$y^R(x^k) \neq y(x^k) \tag{6}$$

for either $k = 0$ or $k = 1$. Substituting the above value of $u^R(x^1)$ into Eq. (5) yields

$$y^R(x^1) = f(x^1, \overline{u}(x^1))$$

which is not equal to $y(x^1) = f(x^1, u(x^1))$. Thus by Eq. (6) any type I type error can be detected if x^k exist such that (1) and (2) are true for $k = 0$ and 1. In a similar fashion it can be shown that type II and III errors will be detected and sufficiency has been shown.

It is evident that condition (1) is necessary. By Fig. 3 for a type I error in u

$$u^R(x^0) = u(x^0)$$
$$u^R(x^1) = \overline{u}(x^1)$$

and for a type II type error in u

$$u^R(x^0) = \overline{u}(x^0)$$
$$u^R(x^1) = u(x^1)$$

Hence for a type I type error in $\overline{u}$

$$y^R(x^0) = f(x^0, \overline{u}(x^0))$$
$$y^R(x^1) = f(x^1, u(x^1))$$

and for a type II error

$$y^R(x^0) = f(x^0, \overline{u}(x^0))$$
$$y^R(x^1) = f(x^1, u(x^1))$$

Thus to detect both a type I and II error it is necessary that both

$$f(x^1, \bar{u}(x^1)) \neq f(x^1, u(x^1))$$

and

$$f(x^0, \bar{u}(x^0)) \neq f(x^0, u(x^0))$$

respectively. Thus the necessity for conditions (1) and (2) is established and the theorem is proved.

Condition (2) of the above theorem is true if and only if

$$\underset{u}{\Delta} f(x, u) = f(x^k, \bar{u}(x^k)) \oplus f(x^k, u(x^k)) = 1 \qquad (7)$$

where $\oplus$ denotes sum, modulo 2, or what is often called the exclusive *or* operation of Boolean algebra. The left side of Eq. (7), the expression $f(x, \bar{u}(x)) \oplus f(x, u(x))$, is called the partial Boolean difference with respect to u and is denoted by the suggestive notation, $\underset{u}{\Delta} f(x, u)$. The partial Boolean difference was first defined and used by the author (Ref. 2) in analyzing certain error-correcting codes. Aker's in Ref. 1 develops some of the calculus and Boolean functional analysis associated with the partial Boolean difference.

In terms of the partial Boolean difference, Theorem 1 can be restated as follows:

Theorem 2. *Let $y = f(x, u(x))$ be a switching function dependent on a gate function $u(x)$ where $x = (x_1, x_2, \cdots, x_n)$. Under assumptions A and B a single fault u of any of types I, II, or III of Fig. 3 can be detected if and only if there exists two input test configurations x^0 and x^1 in $\mathbf{2}^n$ such that*

(1) $u(x^k) = k$

(2) $\underset{u}{\Delta} f(x^k, u(x^k)) = 1$

for both $k = 0$ and $k = 1$.

Section III will be devoted to the development of a difference calculus for Boolean algebra which is quite analogous to the classical difference or differential calculus of many real variables. The chain rules of this calculus simplify the computation of the partial difference required in Theorem 2, making possible the automatic generation of test configurations for all gates of a circuit. As stated, Theorem 2 formalizes rather sufficiently a number of related developments by other authors (Refs. 3–5).

III. Boolean Difference Calculus

To develop a calculus for the Boolean difference, recall first the fundamental expansion formula of Boolean algebra. If $x \epsilon \mathbf{2}$ and $f(x)$ is the mapping $f \mid \mathbf{2} \rightarrow \mathbf{2}$, then

$$\begin{aligned} f(x) &= f(0)\bar{x} \oplus f(1)x \qquad (8) \\ &= f(0) \oplus [f(1) \oplus f(0)]x \\ &= f(0) \oplus \Delta f \mid_{x=0} \cdot x \end{aligned}$$

where $\Delta f \mid_{x=0}$ denotes the difference with respect to x, evaluated at $x = 0$. The second and third forms of Eq. (8) can be recognized as Newton's interpolation formula over the field of two elements, e.g., Ref. 6, pp. 66–70.

The partial Boolean difference was defined by the left side of Eq. (7) in Section II. We now determine some elementary rules and properties of the Boolean difference. Suppose $f(u)$ and $g(u)$ are mappings of $\mathbf{2}$ into $\mathbf{2}$.

Rule I. *$\underset{u}{\Delta} f(u) = f(1) \oplus f(0)$*

For this rule use definition (7) and formula (8). $\underset{u}{\Delta} f = f(u) \oplus f(\bar{u}), f(u) = f(0)u \oplus f(1)\bar{u}$. The rule follows from $u \oplus \bar{u} = 1$.

Rule II. *Δ is a "linear" operator.*

$$\underset{u}{\Delta}(f \oplus g) = \underset{u}{\Delta} f \oplus \underset{u}{\Delta} g$$

This assertion is immediate by definition and the associativity of addition, modulo 2.

Rule III. *If $a \epsilon \mathbf{2}$ is a constant with respect to variable u, $\underset{u}{\Delta} a = 0$.*

Rule IV. $\underset{u}{\Delta}\bar{f} \cdot g = (\underset{u}{\Delta} f) \cdot g \oplus f \cdot (\underset{u}{\Delta} g) \oplus (\underset{u}{\Delta} f) \cdot (\underset{u}{\Delta} g)$.

Rule III is immediate and Rule IV can be verified directly by calculation. Rule IV can be generalized inductively to a product of n terms.

The following rules of differencing are direct consequences of Rules I through IV.

Rule V. $\underset{u}{\Delta} f = \underset{u}{\Delta} f$

This follows from Rules II and III.

Rule VI. $\underset{u}{\Delta} f = \underset{u}{\Delta} f$

Rule VII. *Let $u_1, u_2, \cdots, u_n$ be n binary variables, i.e., variables with domain 2, then*

$$\underset{u_k}{\Delta}(u_1 \cdot u_2 \cdot u_3 \cdots u_n) = u_1 \cdot u_2 \cdots u_{k-1} \cdot u_{k+1} \cdot u_{k+2} \cdots u_n$$

for $k = 1, 2, \cdots, n$.

Rules VI and VII are immediate by Rule I.

An important tool for computing the partial differences, needed in Theorem 2, is provided by the partial difference "chain rule" of the next theorem.

Theorem 3. *Let $f, g_1, g_2, \cdots, g_k$ and u be Boolean functions of n binary variables $x_1, x_2, \cdots, x_n$. Suppose function f is dependent on functions $g_1, g_2, \cdots, g_k$, and in turn functions $g_1, g_2, \cdots, g_k$ are dependent on function u.* Then the partial Boolean difference with respect to u satisfies the chain rule,

$$\begin{aligned}\underset{u}{\Delta} f = {} & \underset{g_1}{\Delta} f \cdot \underset{u}{\Delta} f \oplus \underset{g_2}{\Delta} f \cdot \underset{u}{\Delta} g_2 \oplus \cdots \oplus \underset{g_k}{\Delta} f \cdot \underset{u}{\Delta} g_k \\ & \oplus \underset{g_1 g_2}{\Delta^{(2)}} f \cdot \underset{u}{\Delta} g_1 \cdot \underset{u}{\Delta} g_2 \oplus \underset{g_1 g_3}{\Delta^{(2)}} f \cdot \underset{u}{\Delta} g_1 \cdot \underset{u}{\Delta} g_3 \oplus \cdots \\ & \oplus \underset{g_{k-1} g_k}{\Delta^{(2)}} f \cdot \underset{u}{\Delta} g_k \cdot \underset{u}{\Delta} g_k \oplus \cdots \\ & \oplus \underset{g_1, g_2 \cdots g_k}{\Delta^{(k)}} f \cdot \underset{u}{\Delta} g_1 \cdot \underset{u}{\Delta} g_2 \cdots \underset{u}{\Delta} g_k\end{aligned}$$

where

$$\underset{g_{i_1}, g_{i_2} \cdots g_{i_m}}{\Delta^{(m)}}$$

denotes the mth partial difference with respect to functions $g_{i_1}, g_{i_2} \cdots g_{i_m}$.

Proof: A proof for this theorem is provided here only for $k = 2$. A similar proof can be developed for arbitrary k. For more details see Ref. 5.

For this case f is functionally of the form $f = f(g_1(u), g_2(u))$. Use Eq. (8) to expand $f(g_1, g_2)$ with respect to each variable, separately, as follows:

$$\begin{aligned} f(0, g_2) &= f(0, g_2) \oplus [f(1, g_2) \oplus f(0, g_2)]\, g_1 \\ f(0, g_2) &= f(0,0) \oplus [f(0,1) \oplus f(0,0)]\, g_2 \\ f(1, g_2) &= f(1,0) \oplus [f(1,1) \oplus f(1,0)]\, g_2 \end{aligned}$$

Substituting the second and third equations into the first, yields the formula

$$\begin{aligned} f(g_1, g_2) = {} & f(0,0) \oplus \underset{g_1}{\Delta} f \big|_{g_2=0} \cdot g_1 \\ & \oplus \underset{g_2}{\Delta} f \big|_{g_1=0} \cdot g_2 \oplus \underset{g_1, g_2}{\Delta^{(2)}} f \cdot g_1 g_2 \end{aligned} \tag{9}$$

Formula (9) can evidently be generalized to k variables (Ref. 2).

Take the difference of both sides of Eq. (9) with respect to u, then

$$\underset{u}{\Delta} f = \underset{g_1}{\Delta} f \big|_{g_2=0} \cdot \underset{u}{\Delta} g_1 \oplus \underset{g_2}{\Delta} f \big|_{g_1=0} \cdot \underset{u}{\Delta} g_2 \oplus \underset{g_1 g_2}{\Delta^{(2)}} f \cdot \underset{u}{\Delta}(g_1 g_2) \tag{10}$$

since $f(0,0)$ and the coefficients of g_1, g_2 and $g_1 g_2$ are independent of u. But by Rule IV

$$\underset{u}{\Delta}(g_1 g_2) = (\underset{u}{\Delta} g_1) \cdot g_2 \oplus g_1 \cdot (\underset{u}{\Delta} g_2) \oplus (\underset{u}{\Delta} g_1)(\underset{u}{\Delta} g_2)$$

Substituting this in Eq. (10) and collecting coefficients, we obtain

$$\begin{aligned} \underset{u}{\Delta} f = {} & \big(\underset{g_1}{\Delta} f \big|_{g_2=0} \oplus \underset{g_1, g_2}{\Delta^{(2)}} f \cdot g_2\big) \Delta g_1 \\ & \oplus \big(\underset{g_2}{\Delta} f \big|_{g_1=0} \oplus \underset{g_1 g_2}{\Delta^{(2)}} f \cdot g_1\big) \Delta g_2 \oplus \underset{g_1, g_2}{\Delta^{(2)}} f \cdot \Delta g_1 \cdot \Delta g_2 \end{aligned}$$

The theorem for $k = 2$ is proved once expansion (8) is used to identify the coefficients of Δg_1 and Δg_2 with $\underset{g_1}{\Delta} f$ and $\underset{g_2}{\Delta} f$, respectively.

The above theorem and previous rules provide an adequate machinery for calculating partial differences. In the classical difference calculus there are existence theorems associated with certain classes of difference equations and boundary conditions. Similar theorems can be proved for the present difference calculus of Boolean algebra. An example of such a theorem is the next one. This theorem provides a "global" criterion for the existence of test configurations x^0 and x^1 which satisfy both the "boundary" condition

(1) $u(x^k) = k$

and the difference equation

(2) $\underset{u}{\Delta} f(x, u(x)) = 1$

of Theorem 2.

Theorem 4. *There exist two test configurations x^0 and x^1, satisfying (1) and (2) of Theorem 2 if and only if functions $u(x)$ and $f(x, u(x))$ are such that*

(3) $\overline{u(x)} \cdot \underset{u}{\Delta} f(x, u) \neq 0$

(4) $u(x) \cdot \underset{u}{\Delta} f(x, u) \neq 0$.

Proof: If (1) and (2) of Theorem 2 are true, then $\overline{u(x^0)}\, \underset{u}{\Delta} f(x^0, u) = 1$ and $u(x^1)\, \underset{u}{\Delta} f(x^1, u) = 1$ and (3) and (4) hold. Conversely if (3) is true, there exists x^0 such that $\overline{u(x^0)} \cdot \Delta_u f(x^0, u) = 1$. But this implies (1) of Theorem 2. Similarly (4) implies (2) of Theorem 2 and theorem is proved.

IV. Example and Remarks

Let f be the switching circuit of Fig. 1. f is a circuit of four primary input variables x_1, x_2, x_3, and x_4 with circuit equations given by Eq. (1).

As an application of Theorem 2 we will find test configurations $x^k = (x_1^k, x_2^k, x_3^k, x_4^k)$ for $k = 0, 1$ which will test for a fault in gate u. To do this we must compute the partial difference $\Delta_u f$ and set it equal to one. This is done using the chain rule of Theorem 3 and the difference calculus rules of the last section as follows:

$$\Delta_u f = \Delta_u g \cdot \Delta_g f,$$

$$\Delta_g f = \Delta_{g_1} f \cdot \Delta_g g_1 \oplus \Delta_{g_2} f \cdot \Delta_g g_2 \oplus \Delta^{(2)}_{g_1, g_2} f \cdot \Delta g_1 \cdot \Delta g_2$$

$$\Delta_{g_1} f = g_2, \qquad \Delta_{g_2} f = g_1, \qquad \Delta_{g_1 g_2} = 1$$

$$\Delta_g g_1 = x_1, \qquad \Delta_g g_2 = x_4$$

$$\Delta_u g = x_1 x_4$$

Thus we want to find x^k for $x = 0, 1$ such that

$$u(x^k) = k$$

$$\Delta_u f = \Delta_u g \cdot \Delta_g f = 1$$

where

$$\Delta_u g = x_1 x_4$$

$$\Delta_g f = g_2 \cdot x_1 \oplus g_1 x_2 \oplus x_1 x_4$$

This implies $x^k = (x_1^k, x_2^k, x_3^k, x_4^k)$ must satisfy

$$u(x^k) = k \tag{11}$$

$$\Delta_u g = x_1 x_4 = 1$$

$$\Delta_g f = g_2 x_1 \oplus g_1 x_2 \oplus x_1 x_4 = 1$$

These two configurations are displayed in two rows of the incompleted Table 1. Note that relation $x_1 x_4 = 1$ and $u(x^k) = k$ for $k = 0, 1$ are already satisfied in Table 1. It remains to complete the table, using Eqs. (1), and to show that $\Delta_g f = 1$ can be satisfied.

Table 1. Incompleted table of test configurations for gate u

x_1	x_2	x_3	x_4	u	g	g_1	g_2	f
1			1	0				
1			1	1				

Table 2. Test configurations for gate u ($x_2 x_3 = 0$)

x_1	x_2	x_3	x_4	u	g	g_1	g_2	f
1	1	1	1	0	1	0	0	1
1	—	—	1	1	0	1	1	0

A completed table of test configurations for gate u is shown in Table 2. The two blanks in the table may be filled with elements of the set

$$\{(x_2, x_3) \mid x_2 x_3 = 0\}$$

The solutions of conditions (1) and (2) are not always unique. For this example the test configuration,

$$x^0 = (x_1^0, x_2^0, x_3^0, x_4^0) = (1, 1, 1, 1)$$

is unique and the configuration x^1 is not unique and can be chosen to be

$$x^1 = (x_1^1, x_2^1, x_3^1, x_4^1) = (1, 0, 0, 1)$$

To test for the presence for a fault in any primary input, internal gate, or output one applies Theorem 2 repeatedly, finding test configurations for all such variables. A minimal set of test configurations is obtained by taking minimal union over the set of all test configurations. If n is the number of variables to be tested, the minimal set never exceeds $2n$ test configurations.

References

1. Akers, S. B., Jr., "On the Theory of Boolean Functions," *J. SIAM*, Vol. 2, No. 4, Dec. 1959.
2. Reed, I. S., "A Class of Multiple-Error-Correcting Codes and the Decoding Scheme," *IRE Trans. PGIT-4*, 1954.
3. Armstrong, D. B., "On Finding a Minimal Set of Fault Detection Tests for Combinational Logic Nets," *IEEE Trans. Elec. Comp.*, Vol. IC-15, No. 1, Feb. 1966.
4. Sellers, F. F., Hsiao, M. Y., and Bearnson, L. W., "Analyzing Errors with the Boolean Difference," Annual IEEE Computer Conference, Chicago, Ill., Sept. 1967.
5. Chiang, A. C. L., Reed, I. S., and Banes, A. V., "Data Sensitation, Partial Boolean Difference and Automated Fault Diagnosis," to be published *IEEE Trans. on Elec. Comp.*
6. Van der Waerden, B. L., "Modern Algebra," Vol. I, Frederick Unger Publishing Co., New York, 1949.